

Article

Task-Adaptive Reinforcement Learning for Unified OCR in Maritime and Complex Documents

Yiwen Gao ^{1,*}, Yingqi Zou ^{2,†}, Xiangyu Wang ¹ and Shaohan Wang ¹ 

¹ COSCO SHIPPING Technology Co., Ltd., Shanghai 200135, China; wang.xiangyu@coscoshipping.com (X.W.); wang.shaohan@coscoshipping.com (S.W.)

² School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; zouyingqi@dlmu.edu.cn

* Correspondence: gao.yiwen@coscoshipping.com

† These authors contributed equally to this work.

Abstract

Maritime logistics relies on documents that combine dense multilingual text, irregular layouts, and task-specific output schemas, requiring operational systems to recover text, structure, location, and semantics. We trained Hi-Dolphin-OCR using a task-adaptive reinforcement-learning framework that formulates text spotting, structured localization, document parsing, information extraction, and visual question answering as a unified instruction-following problem. The framework combines task-routed verifiable rewards with bidirectional spatial coverage for spotting, a hard degeneration gate for repetitive, symbol-dominated, and overlong responses, and difficulty-aware sampling based on the supervised model's error distribution. To our knowledge, we also release the first publicly available maritime-oriented multi-task OCR dataset, Maritime-OCR-Bench, comprising 1888 maritime and complex-document examples across five evaluation tasks. On this benchmark, the failure-aware aggregate score increases from 0.7589 for the supervised checkpoint to 0.7900 after reinforcement learning, with the largest differences observed for coordinate-bearing structured outputs. These results support task-aligned reinforcement learning as a practical strategy for unified OCR, while the released dataset provides a reproducible basis for evaluating spatial, structural, and semantic document understanding in maritime-oriented settings.

Keywords: maritime datasets; optical character recognition; multimodal large language model; reinforcement learning; group relative policy optimization



Academic Editor: Andrea Asperti

Received: 2 September 2026

Revised: 28 September 2026

Accepted: 9 October 2026

Published: 12 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Maritime operations depend on large collections of unstructured and semi-structured documents. Bills of lading, packing lists, port forms, container markings, and ship equipment manuals carry operational information about cargo, vessels, ports, schedules, and costs. These documents enter booking, transport, customs, delivery, and settlement workflows. They may be electronic PDF files, scans, mobile photographs, or screenshots. Perspective distortion, rotation, stamps, low resolution, small text, and complex tables often occur together.

The required output also varies across workflows. Container markings require text and spatial coordinates. Bills of lading and cargo forms require strict JSON for downstream entry and validation. Equipment manuals require paragraphs, tables, and reading order for retrieval and question answering. These outputs impose distinct constraints on text,

location, structure, and semantics. Unified document processing therefore requires capabilities beyond sequential text generation. Multi-stage pipelines may also accumulate errors across detection, recognition, layout analysis, and field extraction.

Vision-language models offer a unified image-instruction-response interface for multi-modal tasks [1,2]. Image-to-text generation has also been used to unify document classification, extraction, question answering, and structured parsing [3–5]. Under this interface, an image, an instruction, and a target response can represent text spotting, document parsing, information extraction, or visual question answering. Supervised fine-tuning (SFT) mainly optimizes token-level cross-entropy, while operational document tasks require complete target coverage, parseable JSON, preserved table structure, and semantic equivalence between numerical answers. Task-level optimization signals connect token generation with these output requirements.

The central challenge is to define automatically verifiable optimization signals for heterogeneous document tasks. Spatial, structural, and semantic correctness require complementary reward components. Local or one-sided rewards can create exploitable shortcuts. In prior experiments, a reward that measured only the quality of predicted boxes encouraged the model to emit fewer boxes or blank responses. Format-only rewards produced valid structures with incorrect content. Difficulty-aware allocation directs reinforcement learning capacity toward sources where the SFT model has greater room for improvement.

We address these issues with task-adaptive multi-task OCR reinforcement learning. The method first allocates reinforcement learning samples according to the SFT model's error distribution. A task router then activates only the reward associated with each sample. Structured spotting uses bidirectional coverage to measure both prediction precision and reference completeness. Information extraction, parsing, and visual question answering use key-value, text-structure, and answer-equivalence rewards. A common degeneration gate rejects overlong, repetitive, and symbol-dominated responses before Group Relative Policy Optimization (GRPO) [6].

The main contributions are as follows.

1. We release Maritime-OCR-Bench, a public benchmark for maritime and complex document understanding. It covers structured spotting schemas, parsing, information extraction, and visual question answering under a unified instruction-following format.
2. We introduce task-routed rewards and two safeguards against reward hacking. Bidirectional coverage scores both the prediction and reference sides, thereby penalizing unmatched predictions and missed references. A hard degeneration gate prevents clearly abnormal generations from receiving positive rewards.
3. We develop a difficulty-aware data allocation workflow and a failure-aware evaluation protocol. On the complete benchmark, the recorded aggregate score is 0.0311 higher than that of the matched SFT checkpoint under the same evaluation formulation.

2. Related Work

2.1. Vision-Language Models

Visual instruction tuning established a general image-instruction-response interface for multimodal dialogue and reasoning [1]. Subsequent models expanded either transfer breadth or alignment capacity. PaliGemma couples a SigLIP encoder with Gemma and is designed for transfer across heterogeneous vision-language tasks [7], whereas AlignGPT explicitly models different levels of image-text alignment and composes them according to task requirements [8]. InternVL scales the vision foundation model and aligns it with a language model to support broad visual-linguistic tasks, including document and chart understanding [2]. These models establish the shared interface used by multi-task OCR.

Our framework adds verifiable optimization signals for spatial, structural, and semantic correctness.

High-resolution and text-rich images further expose limits in generic visual representations. Monkey emphasizes input resolution and text supervision [9], while Vary expands the visual vocabulary available to a large vision-language model [10]. TextMonkey combines high-resolution processing with mechanisms for dense text perception in OCR-free document understanding [11]. Related research on degraded imagery and fine-structure perception further illustrates the need for robust visual evidence: VNDHR models multiple nighttime haze degradations and combines ℓ_p , weighted ℓ_2 , and total-variation regularization to recover structure-aware illumination and noise-suppressed reflectance, followed by dark-channel dehazing, nonlinear texture enhancement, and component fusion [12]. SFF-PLDN addresses thin, low-pixel-ratio power lines in complex aerial backgrounds through a shape-aware block that strengthens feature propagation along power-line regions and a spatial-channel attention fusion block that adaptively integrates multi-scale features [13]. Together with the document-oriented models above, these studies motivate robust perception of degraded inputs and fine spatial structures; our work further assigns task-aligned reinforcement learning signals to generated OCR outputs.

Recent studies also use large language models for OCR tasks [14,15]. Figure 1 compares the training pipeline with common language and vision-language model workflows. Hi-Dolphin-OCR adds domain document standardization, multi-task schemas, difficulty-aware allocation, and task-specific verifiable rewards.

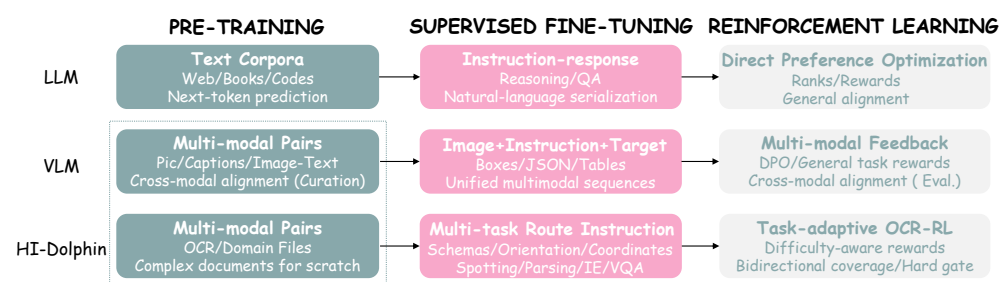


Figure 1. Comparison of training pipelines for text-language models, general vision-language models, and Hi-Dolphin-OCR. All pipelines include pre-training, supervised fine-tuning, and reinforcement learning. Hi-Dolphin-OCR additionally standardizes domain documents, task instructions, page orientation, spatial coordinates, and output schemas. Its reinforcement learning stage uses difficulty-aware allocation and verifiable rewards for text, location, structure, and semantic correctness.

2.2. Document Representation and Generation

Document models differ in where they obtain text and layout evidence. TrOCR uses an image Transformer encoder and an autoregressive text decoder for end-to-end text recognition [16]. LayoutLMv3 retains recognized text and jointly pre-trains text and image representations with unified masking and word-patch alignment [17]. Donut performs OCR-free end-to-end generation of task outputs from document images [3]. Pix2Struct learns visually situated language representations by parsing screenshots into simplified HTML [4], and universal document processing casts multiple document tasks into a common image-to-text formulation [5]. These developments provide a common output interface, while task-specific optimization signals encode heterogeneous notions of correctness.

Recent OCR systems extend this generative formulation from recognition to layout logic and structured document understanding. Logics-Parsing introduces reinforcement learning for complex layout analysis and reading-order prediction [18]; Logics-Parsing-Omni further organizes multimodal parsing from global detection through fine-grained recognition to evidence-grounded explanation [19]. HunyuanOCR uses a compact end-

to-end model for spotting, parsing, information extraction, and visual question answering [20]. Qianfan-OCR generates an explicit layout representation before Markdown output [21], while FireRed-OCR combines a geometry-and-semantics data factory with format-constrained GRPO for document structure parsing [22]. These systems motivate unified OCR generation and reinforcement learning. Our framework routes verifiable rewards by task and couples structured spotting with bidirectional coverage to jointly enforce syntactic validity and reference completeness.

Recent reinforcement-learning approaches use different feedback signals for document and visual tasks. olmOCR 2 trains document OCR with binary unit-test rewards [23], while MonkeyOCR v1.5 uses render-and-compare feedback to assess table structure in document parsing [24]. DocThinker applies rule-based reinforcement learning to document reasoning [25]. GLM-4.5V and GLM-4.1V-Thinking use curriculum sampling in scalable multimodal reinforcement learning across visual recognition, grounding, and document understanding [26].

2.3. Evaluation and Reliability

Evaluation has expanded from transcription accuracy to visual text localization, structured extraction, and reasoning [27,28]. OCRBench v2 groups these capabilities within a broad benchmark and reveals uneven performance across OCR-related tasks [29]. Our benchmark applies this multi-task perspective to maritime-oriented and complex documents, uses two explicit spotting schemas, and reports task-level scores for each OCR capability.

Efficiency and generation reliability remain distinct constraints. V2Drop removes visual tokens with limited internal variation to reduce multimodal inference cost [30], although fine-grained document text and spatial relations make aggressive token removal potentially consequential. Saliency-guided decoding instead rejects visually unsupported token candidates at inference time [31]. Our safeguards operate at training time: the degeneration gate prevents overlong, repetitive, or symbol-dominated outputs from exploiting continuous task rewards, while bidirectional coverage penalizes blank or incomplete spotting predictions.

3. Method

3.1. Problem Formulation

3.1.1. Data Standardization

We represent source documents as image–instruction–answer samples. Each sample retains the task type, reference response, coordinate convention, output schema, and provenance metadata. Only authorized and de-identified material can enter the release candidate. The public evaluation set produced under this protocol is Maritime-OCR-Bench. Its final task distribution is reported in Section 4.1.

The pipeline extracts pages, obtains initial text-layout annotations, isolates unresolved rotations, normalizes coordinates, orders quadrilaterals, converts task schemas, and performs model-assisted and manual quality checks. For an image of width W and height H , each point (x, y) is normalized as $\hat{x} = 1000 \frac{x}{W}$, $\hat{y} = 1000 \frac{y}{H}$. For example, in an image 1000 pixels wide and 2000 pixels high, the pixel coordinates (250, 1000) become (250, 500) on the normalized scale. This operation maps coordinates to a common scale while preserving the image and reference content. Algorithm 1 summarizes the process.

The audited source pool is highly heterogeneous. Figure 2 reports source scale and response-length distributions. The recorded source sizes range from 4250 Grounding examples to 1,000,000 Word VQA examples. Spotting and table parsing contain long response tails, while many Word VQA and formula examples are short. These statistics character-

ize the candidate pool; the final GRPO composition is reported separately. Appendix A (Table A1) summarizes the notation. An input is $x = (I, q, t)$, where I is a document image, q is an instruction, and t is a task type. The reference and generated responses are y^* and y .

Algorithm 1 Multi-task document data standardization

Require: Raw document collection $\mathcal{D}_{\text{raw}}$

Ensure: Standardized releasable set $\mathcal{D}$

```

1:  $\mathcal{D} \leftarrow \emptyset$ 
2: for all  $d \in \mathcal{D}_{\text{raw}}$  do
3:   if  $d$  fails authorization or de-identification checks then
4:     continue
5:   end if
6:   Extract pages and initial text-layout annotations
7:   Detect orientation and isolate unresolved rotations
8:   for all valid page images  $I$  with width  $W$  and height  $H$  do
9:     Normalize coordinates in Section 3.1.1
10:    Order quadrilaterals clockwise from the top-left point
11:    Convert annotations into supported task schemas
12:    Add samples that pass schema, geometry, and quality checks to  $\mathcal{D}$ 
13:  end for
14: end for
15: return  $\mathcal{D}$  with metadata and stable identifiers
  
```

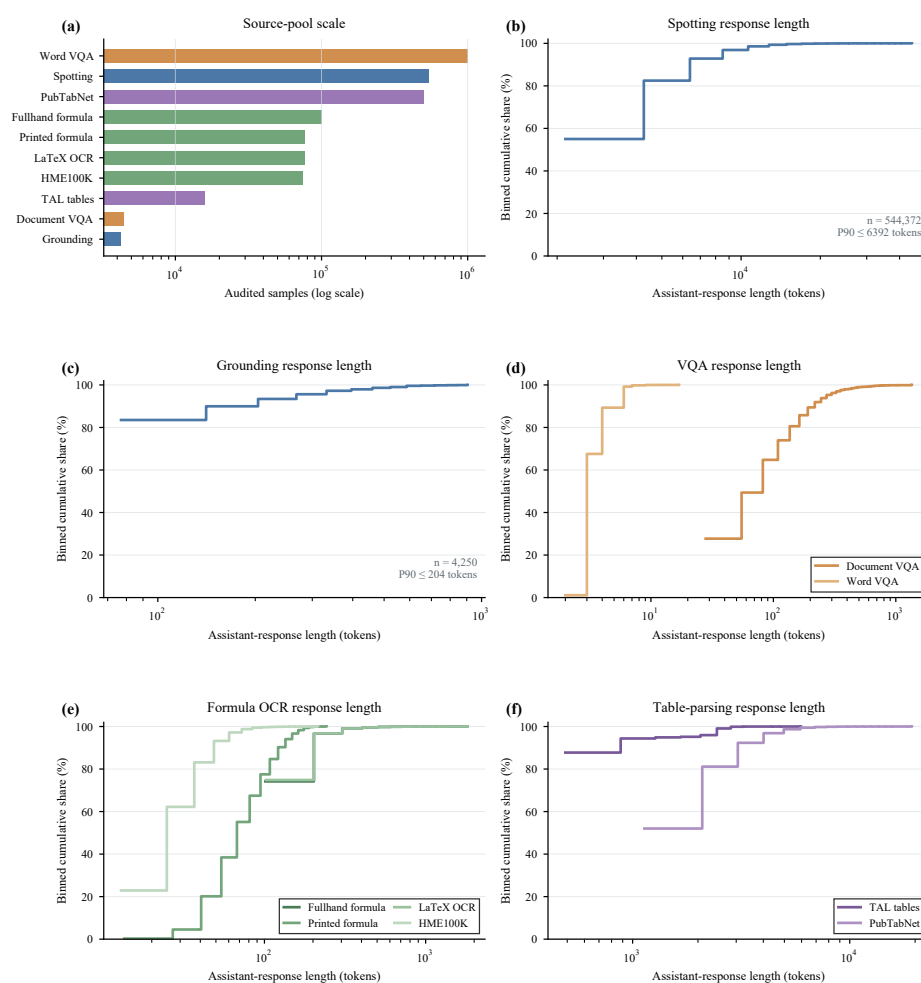


Figure 2. Heterogeneous response complexity in the audited OCR source pool. Panel (a) shows binned sample totals for ten sources on a logarithmic scale. Panels (b–f) show cumulative distributions of

assistant-response token lengths for spotting, grounding, visual question answering, formula recognition, and table parsing. The curves use recorded bin counts and upper bounds. They are not sample-level empirical distributions or final GRPO sampling ratios.

Figure 3 summarizes source records with known counts and distinguishes records included in SFT from excluded records. The figure reports task-record counts, and different task conversions may share a source page. The SFT and GRPO manifests use records with available count fields; ten metadata-only catalogue rows are outside these manifests.

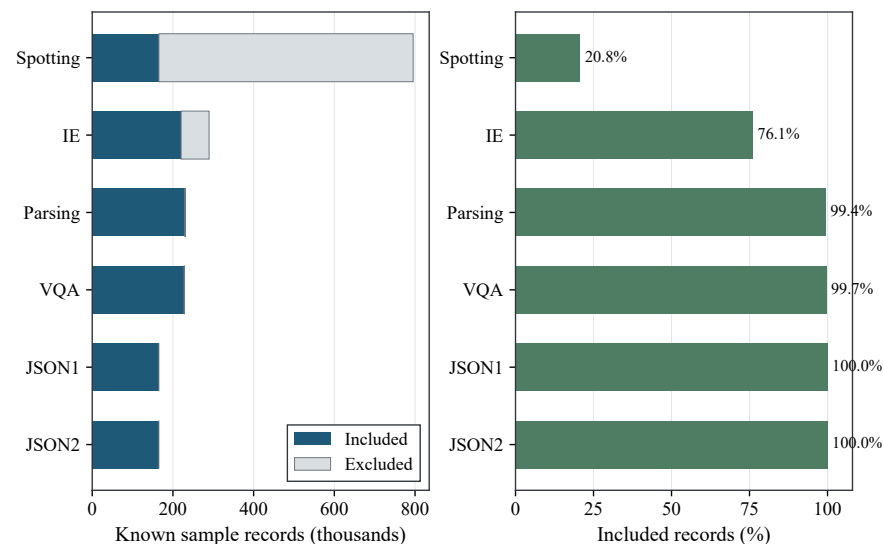


Figure 3. Overview of SFT data selection based on source records with known counts. The left panel reports included and excluded records by task. The right panel reports the corresponding inclusion rates. Counts denote task records, and the training manifests comprise records with available count fields.

3.1.2. Task Definition

We define $\mathcal{T} = \{\text{spotting}, \text{json1}, \text{json2}, \text{parsing}, \text{ie}, \text{vqa}\}$. Spotting returns text and quadrilateral coordinates. JSON1 and JSON2 encode the same localization semantics with two strict schemas. Parsing returns ordered text, Markdown, or tables. Information extraction (IE) returns requested field-value pairs as strict JSON. Visual question answering (VQA) returns a natural-language or numerical answer.

These tasks require four forms of verifiable correctness. Spotting requires content agreement and spatial coverage. IE requires field and value matching. Parsing requires content and structural preservation. VQA requires semantic answer equivalence. JSON validity is recorded as an additional constraint for structured outputs. Each sample retains a stable identifier to support paired model comparisons.

3.2. Overview

Training consists of SFT followed by GRPO. SFT teaches basic recognition and output schemas. GRPO directly optimizes task-level feedback. The old policy samples $G = 4$ responses for each input. A router activates only the relevant reward. Figure 4 shows the complete workflow.

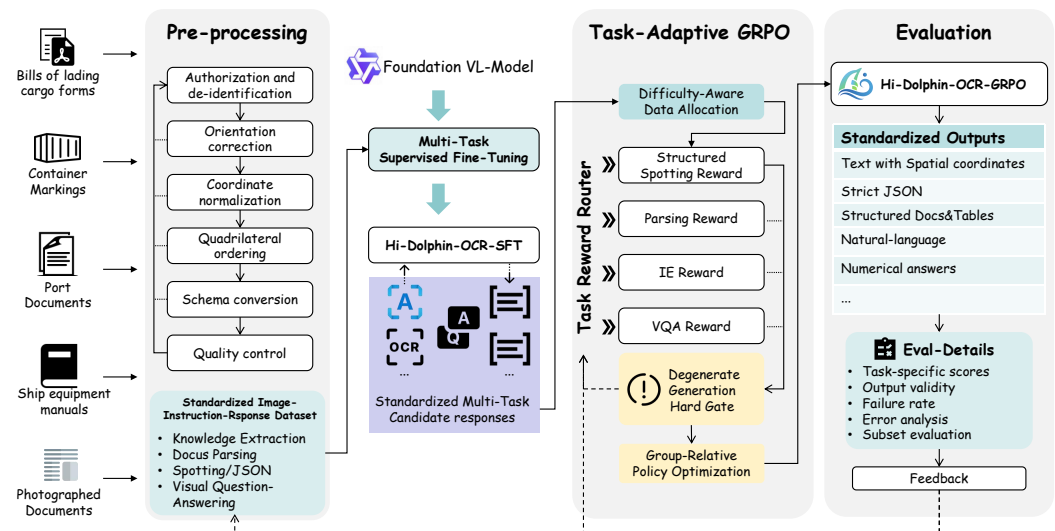


Figure 4. Overview of Hi-Dolphin-OCR. Heterogeneous documents pass through authorization checks, de-identification, orientation correction, coordinate normalization, schema conversion, and quality control. Multi-task SFT provides the initial policy. Difficulty-aware allocation selects GRPO data, and a task router activates the appropriate reward. A hard gate suppresses degenerate responses before policy optimization. The model returns text with coordinates, strict JSON, structured documents, or natural-language answers.

The routing function is

$$R(y, y^*, t) = \begin{cases} R_{\text{spot}}(y, y^*), & t \in \{\text{spotting}, \text{json1}, \text{json2}\}, \\ R_{\text{parse}}(y, y^*), & t = \text{parsing}, \\ R_{\text{ie}}(y, y^*), & t = \text{ie}, \\ R_{\text{vqa}}(y, y^*), & t = \text{vqa}. \end{cases} \quad (1)$$

The sample score contains the active task reward, normalized to $[0, 1]$. Training rewards and offline evaluation metrics are defined separately.

3.3. Text and Parsing Rewards

Let $n(\cdot)$ denote Unicode normalization followed by case folding, punctuation normalization, and whitespace collapse. For two normalized strings $u = n(y)$ and $v = n(y^*)$, E denotes normalized sequence similarity. Compact character F1, F_c , is the multiset character F1 after removing whitespace. The longest-common-subsequence score is $F_l = 2L / (|u| + |v|)$, where L is the character-level longest-common-subsequence length and the score is one when both strings are empty. Lenient similarity E_l additionally removes punctuation and repeated whitespace. Table-aware similarity E_t canonicalizes table delimiters and row separators while retaining cell text and order. All five components lie in $[0, 1]$. Their weighted combination is

$$R_{\text{text}} = 0.25E + 0.20F_c + 0.15F_l + 0.20E_l + 0.20E_t. \quad (2)$$

The implementation subsequently applies limited table-markup compatibility and soft repetition or length-imbalance adjustments before clipping. The hard gate in Section 3.7 remains the final safeguard. Parsing uses the resulting clipped text score,

$$R_{\text{parse}} = \text{clip}(R_{\text{text}}, 0, 1). \quad (3)$$

3.4. Bidirectional Spotting Reward

Let $\mathcal{P}$ and $\mathcal{B}$ denote predicted and reference quadrilaterals. For $p \in \mathcal{P}$ and $b \in \mathcal{B}$, we compute

$$\text{DIoU}(p, b) = \text{IoU}(p, b) - \frac{\|c_p - c_b\|_2^2}{d^2}, \quad (4)$$

where c_p and c_b are box centers and d is the diagonal of their smallest enclosing rectangle. A zero-area quadrilateral or a pair with $d = 0$ is treated as invalid and assigned pair score zero. For all other pairs, the normalized pair score is

$$s(p, b) = \text{clip}\left(\frac{\text{DIoU}(p, b) + 1}{2}, 0, 1\right). \quad (5)$$

Prediction-side coverage measures how well emitted boxes are supported by the reference, while reference-side coverage measures completeness. To define blank and empty-reference cases explicitly, for any two box sets $\mathcal{C}$ and $\mathcal{D}$, we use

$$\text{cov}(\mathcal{C} \rightarrow \mathcal{D}) = \begin{cases} \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \max_{d \in \mathcal{D}} s(c, d), & |\mathcal{C}| > 0, |\mathcal{D}| > 0, \\ 1, & |\mathcal{C}| = |\mathcal{D}| = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

We set $R_{\mathcal{P} \rightarrow \mathcal{B}} = \text{cov}(\mathcal{P} \rightarrow \mathcal{B})$ and $R_{\mathcal{B} \rightarrow \mathcal{P}} = \text{cov}(\mathcal{B} \rightarrow \mathcal{P})$. The layout and final spotting rewards are

$$R_{\text{layout}} = \lambda R_{\mathcal{P} \rightarrow \mathcal{B}} + (1 - \lambda) R_{\mathcal{B} \rightarrow \mathcal{P}}, \quad \lambda = 0.5, \quad (7)$$

$$R_{\text{spot}} = 0.7 R_{\text{layout}} + 0.3 R_{\text{text}}. \quad (8)$$

One-sided coverage allows a model to reduce risk by emitting fewer boxes. Bidirectional coverage adds a reference-side penalty for missed regions. Schema parsing is applied before matching, and malformed structured outputs receive reduced or zero matching scores. Algorithm 2 gives the calculation.

Algorithm 2 Bidirectional structured spotting reward

Require: Candidate y , reference y^* , and task type t

Ensure: Structured spotting reward R_{spot}

- 1: Parse y and y^* under the schema specified by t
 - 2: **if** y cannot provide valid items under the required schema **then**
 - 3: Assign the minimum available matching score
 - 4: **end if**
 - 5: Extract $\mathcal{P}$ and $\mathcal{B}$
 - 6: Compute all pair scores with Equation (5)
 - 7: Compute both coverage directions with Equation (6)
 - 8: Combine spatial coverage with the text reward using Equations (7) and (8)
 - 9: **return** R_{spot}
-

3.5. IE Reward

The IE parser preserves repeated keys. A candidate that cannot be parsed under the required JSON schema receives $R_{\text{ie}} = 0$. For a valid candidate, let A and B be predicted and reference key-value pairs. The functions s_k and s_v are normalized edit similarities computed

with the normalization above on keys and serialized values, respectively. For $a = (k_a, v_a)$ and $b = (k_b, v_b)$,

$$s_{\text{pair}}(a, b) = \begin{cases} 0, & s_k(k_a, k_b) < 0.6, \\ 0.35s_k(k_a, k_b) + 0.65s_v(v_a, v_b), & \text{otherwise.} \end{cases} \quad (9)$$

For the $A \rightarrow B$ direction, pairs are selected greedily without replacement, and the selected scores are averaged over A . The reverse direction exchanges A and B . Empty and unparseable responses are handled before matching. The final reward is

$$R_{\text{ie}} = \frac{1}{2}R_{A \rightarrow B} + \frac{1}{2}R_{B \rightarrow A}. \quad (10)$$

This construction penalizes redundant and missing fields.

3.6. VQA Reward

Let a and b denote the prediction and reference after removing response wrappers. Basic normalization B removes special tokens and coordinate blocks, converts HTML to text, and normalizes Unicode (NFKC), punctuation, case, and whitespace. We use $u = L(B(a))$ and $v = L(B(b))$, where L retains only ASCII letters, ASCII digits, and characters in U+4E00–U+9FFF.

For nonempty u, v , let u_s, u_l denote the shorter and longer strings. The containment score is

$$R_{\text{contain}}(u, v) = \begin{cases} \max\left(0.5, \frac{|u_s|}{|u_l|}\right), & u_s \text{ is a substring of } u_l, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Containment is checked in either direction, with lengths measured in normalized characters. If either string is empty, the containment score is zero.

The digit signature $d(a)$ retains the Unicode decimal digits in $B(a)$ and removes leading ASCII zeros. An all-zero sequence becomes “0”; a sequence without digits has the empty signature ϵ . Let $M_{\text{SM}}(u, v)$ be the total matching-block length returned by Python’s `SequenceMatcher` (`None, u, v`), with its default `autojunk=True`. The prediction is the first sequence, and the reference is the second. For nonempty u, v , soft similarity is

$$s_{\text{soft}}(a, b) = \begin{cases} 1, & u = v \text{ or } d(a) = d(b) \neq \epsilon, \\ \frac{2M_{\text{SM}}(u, v)}{|u| + |v|}, & \text{otherwise.} \end{cases} \quad (12)$$

The matching-block count follows the library algorithm, including its automatic popular-element heuristic. The helper similarity is one for two empty normalized strings and zero when exactly one is empty.

For nonempty u, v , the VQA reward before degeneration gating is

$$R_{\text{vqa}} = \begin{cases} 1, & s_{\text{soft}}(a, b) = 1, \\ R_{\text{contain}}(u, v), & s_{\text{soft}}(a, b) < 1, R_{\text{contain}}(u, v) > 0, \\ 0.5s_{\text{soft}}(a, b), & \text{otherwise.} \end{cases} \quad (13)$$

The implementation then applies the hard gate in Section 3.7 to these nonempty-answer scores. Empty-answer branches return directly: two empty normalized answers receive the default reward 0.3, whereas exactly one empty answer receives zero.

Digit-signature matching provides heuristic tolerance to answer formatting. Because signs, decimal points, units, and surrounding words can be discarded, this heuristic can also conflate distinct answers. For example, “1.2” and “12” receive score one, whereas “10” and “10.0” receive the containment score 2/3 before gating.

3.7. Degeneration Gate

Continuous similarities can assign nonzero scores to abnormal generations. We set $H(y, y^*) = 0$ when any of the following conditions holds: First, the output exceeds three times the reference length and contains at least 120 non-space characters. A semantic character repeats at least 80 times. One semantic token repeats at least 16 times. A phrase of one to four tokens loops at least eight times. A long output contains at least 35% symbols and at least 30 repeated semantic characters. The gate exempts common table-filling characters unless another degeneration criterion is met. We set $H(y, y^*) = 1$ otherwise. The task reward and strict evaluator assign the empty-response score. The gated reward is

$$R'(y, y^*, t) = H(y, y^*)R(y, y^*, t). \quad (14)$$

3.8. Difficulty-Aware Data Allocation

Difficulty-aware allocation is performed once before GRPO. Candidate records are filtered using task-appropriate quality and difficulty indicators, and source-level scores adjust quotas while retaining coverage of each task family. The resulting 49,140-example manifest controls record selection, while reward scales and within-task scoring rules remain fixed. Mini-batches are shuffled from this manifest during the single training epoch. The final task totals are reported in Table 1; the difference from the pilot configuration is documented in Section 4.1.

Table 1. Post-filtering composition of the SFT training split and GRPO manifest. Counts denote task records; one source image may support multiple task conversions.

Stage	Spotting	JSON1	JSON2	IE	Parsing	VQA	Total
SFT	201,911	140,000	100,000	116,784	112,436	141,349	812,480
GRPO	8000	10,000	6000	8000	8000	9140	49,140

3.9. GRPO Objective

For each input, the old policy samples G candidates, denoted by $y_1, \dots, y_G$. The normalized group-relative advantage is

$$\hat{A}_i = \frac{R'_i - \text{mean}(R')}{\text{std}(R') + \delta}, \quad (15)$$

where $\delta > 0$ provides numerical stability. The token-level probability ratio is

$$r_{i,k}(\theta) = \frac{\pi_\theta(y_{i,k} \mid x, y_{i,<k})}{\pi_{\theta_{\text{old}}}(y_{i,k} \mid x, y_{i,<k})}. \quad (16)$$

We optimize

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}[\min(r_{i,k}\hat{A}_i, \text{clip}(r_{i,k}, 1 - \epsilon, 1 + \epsilon)\hat{A}_i) - \beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})]. \quad (17)$$

The KL penalty in Equation (17) is evaluated over sampled completion tokens using the non-negative K3 estimator. For a batch of B sampled completions, the penalty is

$$\hat{D}_{\text{KL}} = \frac{1}{B} \sum_{i=1}^B \frac{1}{T_i} \sum_k m_{i,k} [\exp(\Delta_{i,k}) - \Delta_{i,k} - 1], \quad (18)$$

where $\Delta_{i,k} = \log \pi_{\text{ref}}(y_{i,k} \mid x, y_{i,<k}) - \log \pi_{\theta}(y_{i,k} \mid x, y_{i,<k})$, with x denoting the input associated with completion i . The mask $m_{i,k}$ is one for valid completion tokens and zero for prompt and padding tokens, and $T_i = \sum_k m_{i,k}$ is the number of valid completion tokens. We first average the token penalties within each completion and then average across the B completions, giving each completion equal weight. The reference policy π_{ref} remains frozen throughout GRPO training. Numerical clipping is used in the implementation to stabilize KL-penalty evaluation at extreme probability ratios. We use $G = 4$, $\varepsilon = 0.2$, and $\beta = 0.001$. The complete training loop is provided in Appendix D (Algorithm A1).

4. Experiments

4.1. Datasets

We constructed the training corpus using the standardization protocol described in Section 3.1.1. Quality filtering retained 848,088 candidate task records. Before data splitting, exact image hashes and perceptual hashes with a Hamming-distance threshold of four identified 18,672 within-corpus duplicates and 312 candidates overlapping the test collection, all of which were removed. The resulting corpus contained 829,104 records: 333,427 general-document records (40.2%), 241,140 table and form records (29.1%), 174,112 invoice and receipt records (21.0%), 55,233 scene and dense-text records (6.7%), and 25,192 maritime records (3.0%). We applied a source-grouped train-validation split of 812,480/16,624, ensuring that pages converted into multiple tasks remained within the same partition. A subsequent audit based on source identifiers, image hashes, and nearest perceptual-hash neighbours detected no test pages in either partition.

We used Maritime-OCR-Bench (<https://www.modelscope.cn/datasets/HiDolphin/MaritimeOCRBench/>, accessed on 8 October 2026) as the public test set. The dataset is available on ModelScope under the Apache License 2.0 and contains 1888 examples: 471 IE, 471 VQA, 472 Parsing, 237 JSON1, and 237 JSON2 examples. It is a maritime-oriented mixed benchmark comprising maritime materials, invoices, reports, scans, and scene text.

The task instructions specify the expected output for each record: IE requests JSON key information; Parsing requests Markdown with HTML tables and LaTeX formulas; JSON1 and JSON2 request quadrilateral spotting in distinct JSON schemas; and VQA appends the sample-specific question to an image-grounded instruction. The training corpus additionally uses a tagged text-and-quadrilateral Spotting format. Appendix B records the task prompt templates and output schemas provided in the model interface specification. The decoding settings used in our benchmark are given in Section 4.3.

Figure 5 presents the task distribution. IE, VQA, and Parsing each constitute approximately one quarter of the test set, whereas JSON1 and JSON2 each constitute approximately one eighth.

Table 1 summarizes the post-filtering SFT training split and GRPO manifest. The final GRPO manifest contains 640 more examples than the 48,500-example pilot configuration. These additional examples, together with 500 short instruction-alignment records, are included in the VQA total. Task instances are counted separately when multiple task conversions originate from the same page.

Figure 6 compares the 48,500-example pilot allocation with the final 49,140-example manifest and documents the evolution of the sampling design at the task-quota level.

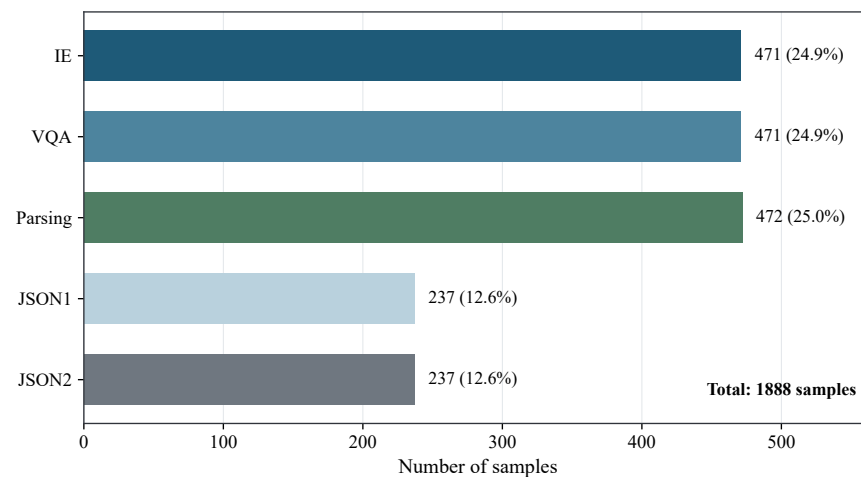


Figure 5. Task distribution of Maritime-OCR-Bench. The benchmark contains 1888 examples: 471 IE, 471 VQA, 472 Parsing, 237 JSON1, and 237 JSON2 examples. Percentages are calculated over the complete test set.

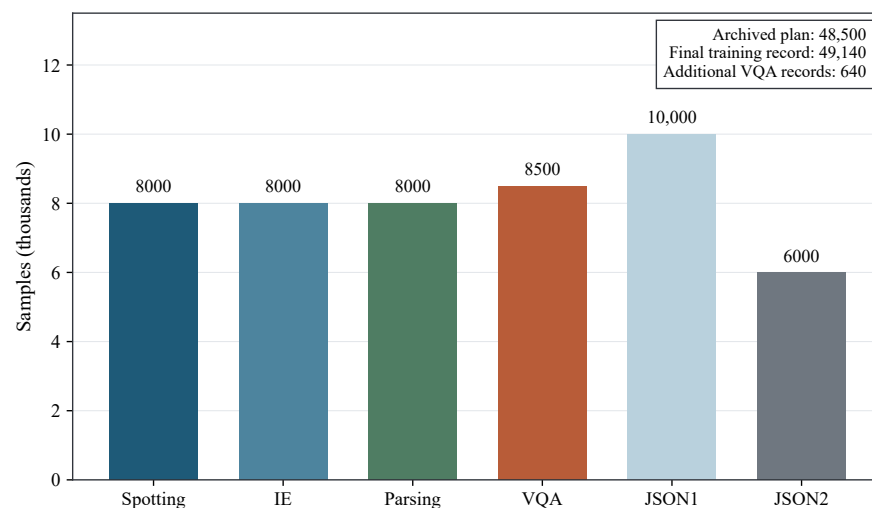


Figure 6. Evolution of the GRPO task-sampling plan. Bars show the 48,500-example pilot quotas, and the annotation indicates the 640-example difference from the final total of 49,140 examples.

4.2. Baselines

We compared Hi-Dolphin-OCR-GRPO with Qwen3-VL-4B-Instruct and the final Hi-Dolphin-OCR-SFT checkpoint. We also evaluated the publicly available HunyuanOCR, Qianfan-OCR, and FireRed-OCR checkpoints [20–22]. All baseline models were evaluated on all five benchmark tasks: IE, VQA, Parsing, JSON1, and JSON2. Overall scores were computed as sample-weighted means over the complete benchmark, with failed generations assigned zero. All models received fixed task prompts, identical rendered images, and the decoding protocol specified below.

4.3. Training Setup

We trained the final model using full-parameter GRPO on two nodes equipped with 16 PPU-ZW810E accelerators. Training comprised one epoch and 3071 complete iterations with a learning rate of 1×10^{-6} . Each global rollout batch contained 64 completions, corresponding to 16 inputs and four candidates per input. Consequently, 49,136 records contributed to parameter updates, while the remaining four records in the 49,140-example manifest were dropped. Both the maximum prompt length and maximum completion length were 4096 tokens. We set the sampling temperature to 1.0 and the KL coefficient

to 0.001. Training required 2 days, 7 h, and 53 min. At the final iteration, the reward was 0.7915, and the corresponding KL divergence was 0.0177.

All experiments were conducted using Ubuntu 22.04, Python 3.10.14, PyTorch 2.5.1, vLLM 0.7.3, Megatron-LM 0.12.1, and ms-swift 3.4.1. We used seed 20260518 for data selection, parameter initialization, and rollout sampling. The resulting checkpoint was used for the evaluation reported in Table 2.

Table 2. Main results on Maritime-OCR-Bench where five-task results are reported for the remaining models. Bold indicates the optimal value.

Model	Overall	IE	VQA	Parsing	JSON1	JSON2	Failure Rate
Qwen3-VL-4B-Instruct	0.6930	0.6742	0.7426	0.7069	0.6328	0.6643	3.34%
HunyuanOCR	0.7284	0.7169	0.7338	0.7544	0.6951	0.7217	2.17%
Qianfan-OCR	0.4594	0.1894	0.1186	0.7821	0.7314	0.7586	1.69%
FireRed-OCR	0.4631	0.2204	0.0739	0.7936	0.7529	0.7708	1.27%
Hi-Dolphin-OCR-SFT	0.7589	0.7488	0.7589	0.8017	0.7002	0.7526	1.91%
Hi-Dolphin-OCR-GRPO	0.7900	0.7762	0.7800	0.8225	0.7603	0.8023	0.64%

For benchmark evaluation, all models received the same task-specific instructions and images, capped at 4000 visual tokens where their interfaces permitted. We used bfloat16 inference and greedy decoding with temperature 0, $\text{top-}p = 1$, a maximum completion length of 4096 tokens, and a batch size of 32. Malformed or empty responses were assigned a score of zero in the failure-aware analysis. Each checkpoint was evaluated once using deterministic decoding.

Figure 7 presents the GRPO training-objective trajectory. Raw rewards fluctuated during early optimization and subsequently became more stable, while a 41-point moving average summarizes the overall trajectory. Test-set performance is reported separately in Section 5.1.

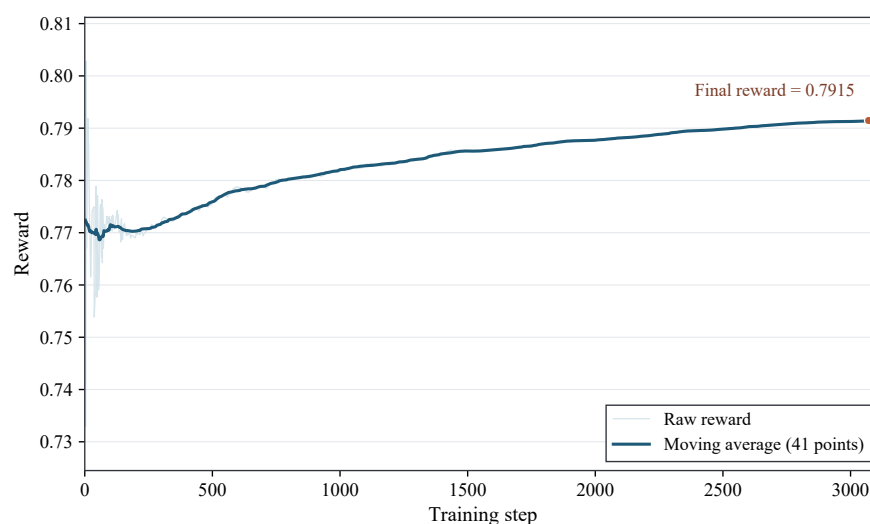


Figure 7. GRPO training reward over optimization steps. The light curve denotes raw rewards, and the dark curve denotes a 41-point moving average. The marker indicates the final reward. Duplicate steps are averaged after sorting; test-set results are reported in Table 2.

4.4. Evaluation Metrics

For offline evaluation, Parsing and VQA were assessed using the text score defined in Equation (2). For a nonempty IE response that could be parsed under the required schema, the task score combined reference-text coverage with strict JSON validity:

$S_{ie} = (\text{coverage} + S_{\text{json}})/2$. Empty or unparseable IE responses were assigned zero before this task-specific formula was applied. JSON1 and JSON2 were evaluated as

$$S_{\text{jsonspot}} = 0.7S_{\text{DIOU}} + 0.3S_{\text{text}}. \quad (19)$$

In addition to task-level scores, we measured JSON parsing success, blank-output rate, predicted-to-reference box-count ratio, reference coverage, overlength rate, and repetition rate. The primary aggregate metric was the strict mean $S_{\text{strict}} = \frac{N_{\text{succ}}S_{\text{succ}}}{N_{\text{all}}}$, which assigns zero to each failed inference. For comparability with the initial benchmark release, we also report the compatibility score. This metric first averages successful generations within each source and then averages the source-level means. All model comparisons and statistical tests instead use the sample-weighted S_{strict} .

We aligned SFT and GRPO predictions using stable sample identifiers for paired statistical analysis. Two-sided 95% percentile confidence intervals were estimated from 10,000 task-stratified paired bootstrap replicates using seed 20260820. We additionally performed two-sided paired permutation tests with 1,000,000 sign exchanges and applied Holm correction across the five task-level comparisons. These analyses quantify sampling uncertainty over the evaluated examples; multi-seed ablations separately characterize training variation for the reward variants.

5. Results

5.1. Main Results

Table 2 summarizes the available failure-aware evaluation records, with missing, empty, or unparseable responses assigned zero. Under this formulation, Hi-Dolphin-OCR-GRPO records 0.7900, compared with 0.7589 for its SFT initialization and 0.6930 for Qwen3-VL-4B-Instruct. The original compatibility evaluator, which omitted failed generations and used source-balanced aggregation, reported 0.7941 overall. Its successful-generation task scores were 0.7795, 0.7817, 0.8260, 0.7700, and 0.8161 for IE, VQA, Parsing, JSON1, and JSON2, respectively. Twelve GRPO generations failed: 2 IE, 1 VQA, 2 Parsing, 3 JSON1, and 4 JSON2 examples. After assigning zero to these failures and weighting samples uniformly, the recorded overall score is 0.7900. The matched SFT-to-GRPO records support the paired analysis in Table 3; external-system aggregates provide descriptive comparisons under the available records.

Table 3 records the reported paired comparison. The task-level differences are positive and are largest for JSON1 and JSON2. The confidence intervals and adjusted p values summarize the stated resampling analysis.

Table 3. Reported matched SFT-to-GRPO changes on Maritime-OCR-Bench. Confidence intervals use 10,000 task-stratified paired bootstrap replicates. p values are from two-sided paired permutation tests. The five task-level values use Holm correction; the overall comparison is reported separately.

Task	Samples	SFT	GRPO	Change	95% CI	Adjusted p
IE	471	0.7488	0.7762	+0.0274	[0.0171, 0.0379]	0.0002
VQA	471	0.7589	0.7800	+0.0211	[0.0124, 0.0298]	0.0006
Parsing	472	0.8017	0.8225	+0.0208	[0.0129, 0.0287]	$<10^{-5}$
JSON1	237	0.7002	0.7603	+0.0601	[0.0424, 0.0779]	$<10^{-5}$
JSON2	237	0.7526	0.8023	+0.0497	[0.0328, 0.0669]	$<10^{-5}$
Overall	1888	0.7589	0.7900	+0.0311	[0.0248, 0.0374]	$<10^{-5}$

Figure 8 provides broader descriptive context using three external API models and Hi-Dolphin-OCR. The heatmap reports the recorded compatibility scores under each

available external-system configuration. Controlled SFT-to-GRPO evidence is provided by the matched comparison in Table 3.

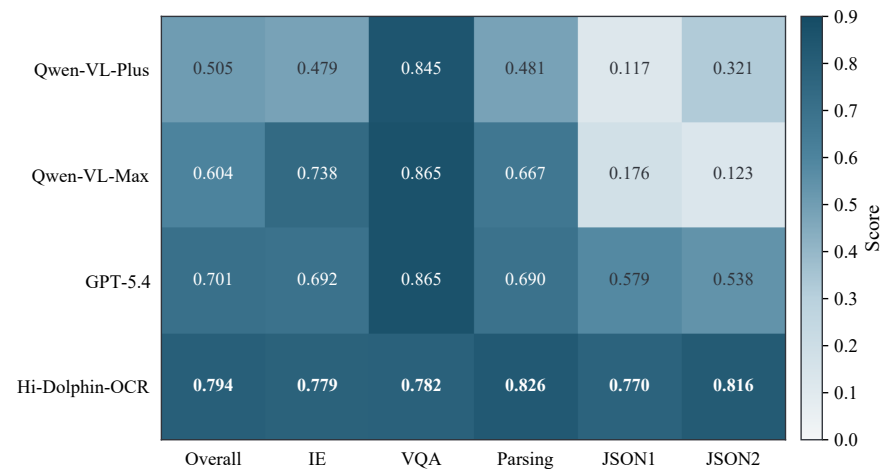


Figure 8. Compatibility-score heatmap for three external API systems and Hi-Dolphin-OCR. Cells show recorded Overall, IE, VQA, Parsing, JSON1, and JSON2 scores under the original successful-generation aggregation and the available system configurations.

5.2. Reward Safeguards

The ablation study compared a format-only reward, a DIOU-text-format reward, one-sided coverage, bidirectional coverage without hard gating, and the full method under uniform or difficulty-aware sampling. All variants were initialized from the same SFT checkpoint and used the same nominal training manifest.

Early development experiments showed that one-sided spotting rewards assessed emitted predictions while leaving reference completeness unscored. This objective can favor sparse outputs because omitted references contribute no penalty to the prediction-side average. Table 4 reports mean performance across three seeds. Bidirectional coverage was associated with higher reference coverage and fewer blank outputs than one-sided coverage, while hard gating reduced degenerate outputs. These comparisons support the combined safeguards; component-level causal estimates require factorial controls with one change per variant.

Table 4. Reported ablations on Maritime-OCR-Bench; values are means over three training seeds, and uncertainty for the strict score is the standard deviation. The first five variants use uniform sampling. Schema validity uses JSON1, JSON2, and IE, whereas spatial diagnostics use JSON1 and JSON2. Degeneration denotes an output satisfying any hard-gate overlength or repetition criterion. Bold indicates the optimal value.

Variant	Strict Score	Schema Valid	Reference Coverage	Box-Count Ratio	Blank	Degeneration
Format only	0.7218 ± 0.0026	98.7%	0.641	0.69	3.82%	0.21%
DIOU + text + format	0.7534 ± 0.0021	96.1%	0.731	0.81	2.56%	0.74%
One-sided coverage	0.7612 ± 0.0024	96.7%	0.748	0.76	5.91%	0.83%
Bidirectional coverage	0.7810 ± 0.0020	96.4%	0.864	0.97	0.88%	2.46%
Bidirectional coverage + gate	0.7837 ± 0.0019	98.2%	0.872	0.98	0.51%	0.33%
Full method (difficulty-aware)	0.7900 ± 0.0019	98.7%	0.884	0.99	0.34%	0.14%

5.3. Development Diagnostics

Development records provide additional evidence about the evolution of the method across different evaluation sizes and training configurations. We report them as configuration-specific diagnostics.

Figure 9 separates evaluations containing 235, 2192, and 3790 text boxes into distinct configuration panels. On the early 235-box set, mean IoU increased from 0.0022 to 0.3155. In the 3790-box records, reinforcement learning increased the mean IoU for version 0227 from 0.3458 to 0.3936. Results across the subsequent 0302 and 0304 configurations varied, indicating that the spatial gains depend on the training and inference configuration. Two 0226 records also used different inference scales.

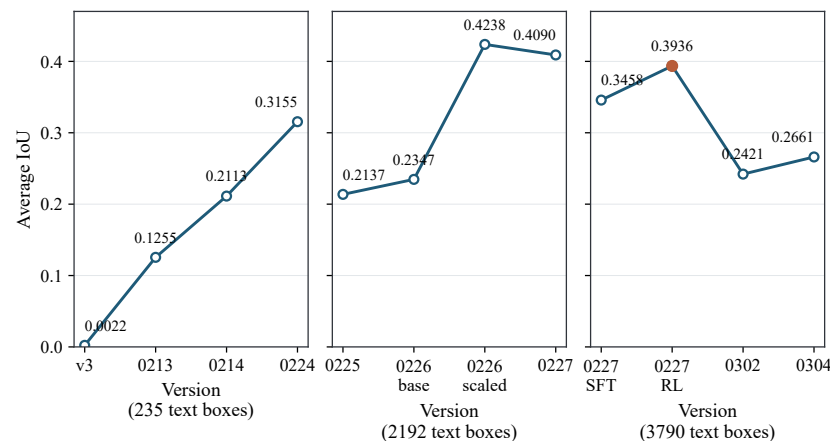


Figure 9. Mean IoU across development versions at three fixed evaluation scales. The panels contain 235, 2192, and 3790 text boxes, respectively. Values across panels are not connected or treated as directly comparable gains. The highlighted point marks the post-reinforcement-learning result for the 0227 version in the 3790-box record.

Figure 10 reports early single-task reward checks. VQA, IE, and Parsing each used 1000 training examples and 62 iterations. The retained initial and final means increased for all three tasks, providing an early confirmation that each reward signal supported effective optimization. Final benchmark performance is reported separately in Section 5.1.

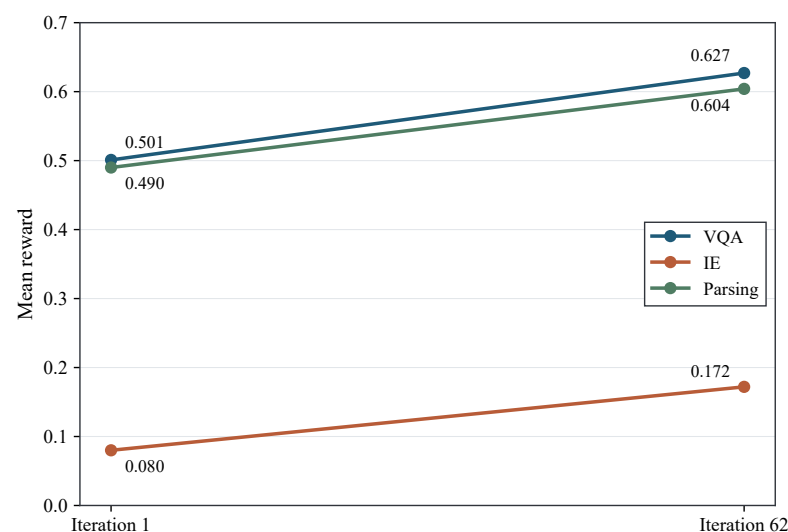


Figure 10. Initial and final mean rewards in early single-task experiments. VQA, IE, and Parsing each used 1000 examples and 62 training iterations. Lines connect the two recorded endpoints for each task.

5.4. Maritime Subsets

Table 5 reports scores for a 200-example maritime subset grouped into four document categories. The recorded differences are positive in all four groups and are largest for container markings. This subset provides domain-specific evidence within the broader mixed benchmark.

Table 5. Reported strict scores on the maritime subset. Changes are absolute GRPO-minus-SFT differences.

Document Type	Samples	SFT	GRPO	Change
Container markings	52	0.7014	0.7695	+0.0681
Bills of lading and transport forms	58	0.7554	0.7992	+0.0438
Port and cargo documents	47	0.7289	0.7826	+0.0537
Ship and machinery manuals	43	0.7837	0.8124	+0.0287
All maritime documents	200	0.7412	0.7904	+0.0492

5.5. Cases Study

Table 6 summarizes the recorded error labels for 612 reviewed examples. The counts are lower after GRPO for most categories, particularly overlength or repetition, invalid schemas, and omissions. Complex-table geometry is the exception. The table characterizes the reviewed sample; population-level rates require probability sampling from the deployment distribution.

Table 6. Reported error counts on 612 reviewed examples. An example may receive more than one label. Relative change is $(N_{\text{GRPO}} - N_{\text{SFT}}) / N_{\text{SFT}}$.

Error Category	SFT	GRPO	Relative Change
Omitted text or boxes	183	112	−38.8%
Invalid JSON or schema	96	41	−57.3%
Redundant boxes or fields	84	52	−38.1%
Spatial displacement	117	83	−29.1%
Small-target recognition	91	73	−19.8%
Perspective distortion	58	51	−12.1%
Handwriting recognition	44	42	−4.5%
Complex-table geometry	47	55	+17.0%
Overlength or repetition	39	6	−84.6%

Figures A1–A4 show four representative inputs and responses as separate, enlarged examples. In the first container example (Figure A1), the model recognizes the identifier and weights but maps capacity fields to inconsistent schema keys. The second container response (Figure A2) matches the visible text; box-level references are required to quantify localization accuracy. The invoice response (Figure A3) recovers visible identifiers, dates, quantities, line items, and totals. The receipt answer in Figure A4 gives 6%, matching the visible tax rate. These examples expose usable outputs together with field-mapping and localization uncertainties, complementing the aggregate evaluation.

6. Discussion

The task rewards are designed as operationally aligned proxies for spatial, structural, and semantic correctness. Bidirectional coverage emphasizes both prediction quality and reference completeness, while independent best matching keeps rollout-time computation tractable. Future extensions may incorporate one-to-one assignment, stricter schemas, or learned calibration to model text–box associations and closely related field names at finer

granularity. The Parsing and Spotting rewards share components with their offline metrics. IE training emphasizes bidirectional field semantics, while offline evaluation emphasizes text coverage and JSON validity. VQA training accepts numerical and containment equivalence, while offline evaluation uses a text composite. These complementary formulations can produce different rankings, so training rewards characterize optimization progress and benchmark scores characterize held-out task performance.

A unified model can reduce orchestration across maritime document processors. IE and Parsing can support field entry, rule checks, and archiving for bills of lading and cargo forms. Structured spotting can locate container identifiers for human verification. Parsing and VQA can support retrieval and question answering over equipment manuals. Each workflow applies a task-specific validation interface to the shared visual representation.

The method is designed for a model-generation, rule-validation, human-review, and feedback workflow. JSON schemas, field ranges, document logic, and cross-field consistency screen model outputs. Human reviewers resolve low-confidence or conflicting cases and approve outputs used in high-risk business decisions.

The present study evaluates one 4B backbone on a maritime-oriented mixed benchmark. Reproduction covers the public benchmark, prompts, metrics, and reported configuration; some private or distilled training sources remain access-controlled. The evidence comprises paired estimates, multi-seed reward-safeguard comparisons, a maritime subset, and reviewed error counts. Factorial ablations are the next step for estimating individual component effects and interactions. Conventional word-error-rate and end-to-end spotting metrics will extend comparison with specialized OCR systems across additional model scales, document domains, languages, and imaging conditions. One-to-one text–layout assignment and annotation-confidence estimates may also improve reward calibration and sample allocation.

7. Conclusions

We presented a task-adaptive reinforcement learning method for multi-task OCR in maritime and complex documents. The method represents spotting, parsing, information extraction, and visual question answering as verifiable generation tasks. A task router connects each task to an appropriate reward. Bidirectional coverage addresses sparse and blank spotting outputs. A common hard gate suppresses overlong and repetitive responses. Difficulty-aware allocation directs the reinforcement learning budget according to the SFT model's error distribution.

In the available failure-aware aggregate records, Hi-Dolphin-OCR-GRPO reaches 0.7900 on the maritime-oriented mixed Maritime-OCR-Bench test set, compared with 0.7589 for its matched SFT checkpoint. The largest descriptive differences occur for the coordinate-bearing JSON tasks. These results support the combined task-routed rewards and coverage safeguards; factorial ablations will quantify individual component effects. The method provides a common interface between operational failure modes, task-level optimization, and machine-checkable outputs within validated human-in-the-loop workflows.

Author Contributions: Y.G. and Y.Z. contributed equally to this work. All authors contributed to the conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, and writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China-Developed Multimodal Large Model and Knowledge Platform for the Maritime Industry (Phase II), grant number YAN5C6C/KY0092.

Data Availability Statement: Our data is available on <https://www.modelscope.cn/datasets/HiDolphin/MaritimeOCRBench/>, accessed on 8 October 2026.

Conflicts of Interest: Authors Yiwen Gao, Xiangyu Wang and Shaohan Wang were employed by COSCO SHIPPING Technology Co., Ltd., Shanghai, China. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Notation

Table A1. Core notation used in the proposed framework.

Symbol	Definition
I, q, t	Document image, task instruction, and task type.
$x = (I, q, t)$	Input to the policy model.
y^*, y	Reference response and generated response.
$\mathcal{T}, \mathcal{D}$	Task set and standardized multi-task dataset.
$\pi_\theta, \pi_{\text{ref}}$	Trainable policy and KL reference policy.
$R(y, y^*, t)$	Task reward selected by the routing function.
$H(y, y^*)$	Hard gate for degenerate generation.
$R'(y, y^*, t)$	Reward after hard gating.
$G, \hat{A}_i$	Number of candidates and normalized group-relative advantage.
$\mathcal{P}, \mathcal{B}$	Predicted and reference box sets.

Appendix B. Task Prompt Templates

English task templates from the Hi-Dolphin-OCR interface specification are listed below. Each request includes an image; VQA includes a sample-specific question.

Spotting. “Detect and recognize all text instances in the image, and output them in reading order from top to bottom and left to right. Each text instance must include the recognized text and its four-point coordinates (top-left, top-right, bottom-right, bottom-left). If the content is a table, output the entire table as a single instance. The table content must be represented using a <table> structure (including <tr> and <td> tags) and must not be split into multiple instances. The output format must strictly follow:”

```
<|object_ref_start|>text<|object_ref_end|>
<|quad_start|>(x1,y1),(x2,y2),(x3,y3),(x4,y4)<|quad_end|>
```

“Do not output any additional content.”

IE. “Extract all key information and return it in JSON format.”

Parsing. “Extract all information from the main body of the document image and represent it in Markdown format. Ignore headers and footers. Represent tables using HTML format and mathematical formulas using LaTeX format. Parse and organize the content according to the natural reading order.”

JSON1. Use the Spotting instruction through the table constraint, followed by “The output must be in JSON format, and only JSON should be returned:”

```
{"image": [
  {"text": "<text>",
   "position": [[x1,y1],[x2,y2],[x3,y3],[x4,y4]]}
]}
```

“Do not output any additional content.”

JSON2. Use the Spotting instruction through the table constraint, followed by “The output must be in JSON format, and only JSON should be returned:”

```
{
  "images": [
    {
      "content": [
        {
          "id": 0,
          "type": "text",
          "text": "<text>",
          "pos": [x1,y1,x2,y2,x3,y3,x4,y4]
        }
      ]
    }
  ]
}
```

“Do not output any additional content.”

VQA. “Please answer the question based on the image:” followed by the record-specific question.

Appendix C. Additional Training Configuration

Final GRPO configuration: micro-batch size 1 per data-parallel process; generation batch 64 completions (16 inputs, 4 candidates each); bfloat16; token-level importance sampling; image-token limit 4000; vLLM context limit 8192 tokens; lower and upper clipping coefficients 0.2. Other training settings are listed in Section 4.3.

Archived SFT script: learning rate 2×10^{-5} ; minimum learning rate 10^{-6} ; warmup fraction 0.05; weight decay 0.1; gradient clipping 1.0; micro/global batch sizes 12/288; maximum sequence length 32,768 tokens; one epoch; frozen vision encoder and aligner.

Archived offline allocation protocol: split candidate responses at the 10th, 20th, ..., 90th token-length percentiles; score sampled responses with task-specific rewards; standardize source-level mean scores as z_s ; allocate source quotas with $w_s \propto \exp(-z_s/\tau)$ subject to source availability. Example allocation temperature: $\tau = 0.7$.

Appendix D. GRPO Training Algorithm

Algorithm A1 Task-adaptive multi-task GRPO training

Require: SFT policy π_θ , π_{ref} , $\mathcal{D}$, group size G , clipping coefficient ε , and KL coefficient β

Ensure: Optimized policy π_θ

```
1: while the training budget is not exhausted do
2:   Set  $\theta_{\text{old}} \leftarrow \theta$  and hold the rollout policy fixed for this iteration
3:   Sample a mini-batch using difficulty-aware data allocation
4:   for all  $(x, y^*)$  in the mini-batch, where  $x = (I, q, t)$  do
5:     Sample  $y_1, \dots, y_G \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ 
6:     for  $i = 1$  to  $G$  do
7:       Select  $R(y_i, y^*, t)$  with Equation (1)
8:        $R'_i \leftarrow H(y_i, y^*)R(y_i, y^*, t)$ 
9:     end for
10:    Compute group-relative advantages with Equation (15)
11:  end for
12:  Update  $\theta$  with the objective in Equation (17)
13: end while
14: return  $\pi_\theta$ 
```

Appendix E. Qualitative Examples



Task Extract all information from the container image.

Model response CONTAINER_NUMBER: DFSU2975344; MAX_GROSS_KGS: 30,480; TARE_KGS: 2186; NET_KGS: 28,295; CU_CAP_KGS: 33.2; CU_CAP_CBM: 1173; CU_CAP_CLFT: 1173.

Assessment Partial: the identifier and weight fields are visually supported, but the capacity keys and values are inconsistently mapped.

Figure A1. Container information extraction example. The model recovers the core identifier and weight fields, but the capacity keys and values are inconsistently mapped.



Task Recognize all text and return quadrilateral coordinates in JSON.

Model response FSCU8844455; 45G1; MAX. GROSS 32,500 KGS / 71,650 LBS; TARE 3830 KGS / 8440 LBS; MAX. CARGO 28,670 KGS / 63,210 LBS; 76.4 CU.M.; 2700 CU.FT.

Assessment Visually consistent: the principal text values are present in the image; coordinate accuracy is not scored without box-level ground truth.

Figure A2. Container text spotting example. The principal recognized text is visually consistent with the input; coordinate references are required for box-level localization scoring.

David Brown Santasalo Soft Information Co., Ltd.
479 Shelton Divide Suite 286 North Isaiahville, FL 17035
Tel: (302) 393-3912 Fax: (310) 962-2235

CONTRACT

Invoice number: 2146103130 Date: Dec.23.2021
Buyer's order No: 4500230691

Order ID: ISPE300534 Kind of package: Plywood box

Buyer:
Soft Information Co., Ltd.
7650 Murphy Vista Liville, UT 26169
USCCG Pugh FPO AP 11710
0722 Sarah Harbor Longtown, WY 26635

Delivery address:
Soft Information Co., Ltd.
Unit 9254 Box 1923 DPO AA 43200
USNS Brown FPO AA 02258
3767 Angela Neck Aliburgh, NY 60519

Payment terms: 30 days net Terms of delivery: EXW Suzhou

From: China To: Indonesia

Gross weight (kg): 94 Net weight (kg): 85
Pack Quantity: 1 Volume (m³): 0.12
Dimensions(cm): 60 x 50 x 40

DESCRIPTION:

NO.	Item ID	Description	QTY (PCS)	UNIT PRICE USD	AMOUNT (USD)
1	IWR0114574	PINION	1	642	642.00
2	IWR0114573	GEAR WHEEL	1	1,020	1,020.00
3	LU6RML170	OIL LEVEL GLASS	1	100	100.00
TOTAL:				USD	1,762.00



Task Extract invoice, logistics, weight, volume, and amount fields as JSON.

Model response Invoice: 2146103130; Date: Dec.23.2021; Order: 4500230691; Gross/Net: 94/85 kg; Volume: 0.12 m³; Item amounts: USD 642.00, 1020.00, 100.00; Total: USD 1762.00.

Assessment Visually consistent: the listed fields and arithmetic total match the document.

Figure A3. Invoice field extraction example. The response recovers the visible identifiers, dates, weights, volume, line-item amounts, and arithmetic total.

MR. D.I.Y. SDN BHD
(CO.REG :704427-T)
LOT 1851-A & 1851-B, JALAN KPB 6,
KAWASAN PERINDUSTRIAN BALAKONG,
43300 SERI KEMBANGAN, SELANGOR
(GST ID No :001092886528)
(CHERAS LEISURE MALL)
-TAX INVOICE-

CLOTH PEG E-140 *S
WA36 - 6/60
9555112901406 1 X 5.90 5.90

Item(s) : 1 Qty(s) : 1

Total Incl. GST@6% RM 5.90
CASH RM 5.90

GST Summary Amt(RM) Tax(RM)
GST S@6% 5.57 0.33

18-10-17 15:16 SH01 ZT09 T3 R000000163
OPERATOR GCHT - NOR MASLIZA

EXCHANGE ARE ALLOWED WITHIN
7 DAYS WITH RECEIPT.
STRICTLY NO CASH REFUND.

Task Question: What is the tax rate on this receipt?

Model response Answer: 6%.

Assessment Correct: the receipt states "GST@6%" and "GST S@6%".

Figure A4. Receipt visual question answering example. The predicted tax rate of 6% agrees with the rate printed on the receipt.

References

1. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 34892–34916. [\[CrossRef\]](#)
2. Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2024; pp. 24185–24198. [\[CrossRef\]](#)
3. Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; Park, S. OCR-Free Document Understanding Transformer. In *Proceedings of the Computer Vision—ECCV 2022*; Springer: Cham, Switzerland, 2022; pp. 498–517. [\[CrossRef\]](#)
4. Lee, K.; Joshi, M.; Turc, I.R.; Hu, H.; Liu, F.; Eisenschlos, J.M.; Khandelwal, U.; Shaw, P.; Chang, M.W.; Toutanova, K. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*; PMLR: Brookline, MA, USA, 2023; pp. 18893–18912.
5. Tang, Z.; Yang, Z.; Wang, G.; Fang, Y.; Liu, Y.; Zhu, C.; Zeng, M.; Zhang, C.; Bansal, M. Unifying Vision, Text, and Layout for Universal Document Processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2023; pp. 19254–19264. [\[CrossRef\]](#)
6. Zhou, H.; Ye, K.; Xu, E.; Zhu, J.; Yang, Y.; Gong, S.; Shi, C. Demystifying Group Relative Policy Optimization: Its Policy Gradient is a U-Statistic. *arXiv* **2026**, arXiv:2603.01162.
7. Beyer, L.; Steiner, A.; Pinto, A.S.; Kolesnikov, A.; Wang, X.; Salz, D.; Neumann, M.; Alabdulmohsin, I.; Tschannen, M.; Bugliarello, E.; et al. Paligemma: A versatile 3b vlm for transfer. *arXiv* **2024**, arXiv:2407.07726.
8. Zhao, F.; Pang, T.; Li, C.; Wu, Z.; Guo, J.; Xing, S.; Dai, X. Alignnpt: Multi-modal large language models with adaptive alignment capability. *arXiv* **2024**, arXiv:2405.14129.
9. Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; Bai, X. Monkey: Image Resolution and Text Label Are Important Things for Large Multi-modal Models. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 17–21 June 2024; pp. 26753–26763. [\[CrossRef\]](#)
10. Wei, H.; Kong, L.; Chen, J.; Zhao, L.; Ge, Z.; Yang, J.; Sun, J.; Han, C.; Zhang, X. Vary: Scaling up the Vision Vocabulary for Large Vision-Language Model. In *Proceedings of the Computer Vision—ECCV 2024*; Springer: Cham, Switzerland, 2024; pp. 408–424. [\[CrossRef\]](#)
11. Liu, Y.; Yang, B.; Liu, Q.; Li, Z.; Ma, Z.; Zhang, S.; Bai, X. TextMonkey: An OCR-Free Large Multimodal Model for Understanding Document. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 6008–6019. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Liu, Y.; Wang, X.; Hu, E.; Wang, A.; Shiri, B.; Lin, W. VNDHR: Variational Single Nighttime Image Dehazing for Enhancing Visibility in Intelligent Transportation Systems via Hybrid Regularization. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 10189–10203. [\[CrossRef\]](#)
13. Zhang, S.; Zhang, X.; Ren, W.; Shen, L.; Zhang, J. Shape-aware and feature fused power line detection network. *Eng. Appl. Artif. Intell.* **2026**, *170*, 113981. [\[CrossRef\]](#)
14. Zheng, Q.; Sun, Y.; Wang, L.; Zhang, H.; Meng, F.; Li, M. FeOCR: Domain-Adaptive Chinese OCR with Visual Character Disambiguation and LLM-Based Correction for Metallurgical Documents. *Electronics* **2026**, *15*, 1144. [\[CrossRef\]](#)
15. Papazis, S.; Giotis, A.P.; Nikou, C. Enhancing Keyword Spotting via NLP-Based Re-Ranking: Leveraging Semantic Relevance Feedback in the Handwritten Domain. *Electronics* **2025**, *14*, 2900. [\[CrossRef\]](#)
16. Li, M.; Lv, T.; Chen, J.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 13094–13102. [\[CrossRef\]](#)
17. Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; Wei, F. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, Lisbon, Portugal, 10–14 October 2022; pp. 4083–4091. [\[CrossRef\]](#)
18. Chen, X.; Li, S.; Zhu, X.; Chen, Y.; Yang, F.; Fang, C.; Qu, L.; Xu, X.; Wei, H.; Wu, M. Logics-parsing technical report. *arXiv* **2025**, arXiv:2509.19760.
19. An, X.; Cai, J.; Chen, X.; Liu, H.; Liu, P.; Wang, P.; Yang, B.; Zhu, X.; Chen, Y.; Gao, Y.; et al. Logics-parsing-omni technical report. *arXiv* **2026**, arXiv:2603.09677.
20. Hunyuan Vision Team; Lyu, P.; Wan, X.; Li, G.; Peng, S.; Wang, W.; Wu, L.; Shen, H.; Zhou, Y.; Tang, C.; et al. HunyuanOCR Technical Report. *arXiv* **2025**, arXiv:2511.19575.
21. Dong, D.; Zheng, M.; Xu, D.; Luo, C.; Zhuang, B.; Li, Y.; He, R.; Wang, H.; Zhang, W.; Wang, W.; et al. Qianfan-OCR: A unified end-to-end model for document intelligence. *arXiv* **2026**, arXiv:2603.13398.
22. Wu, H.; Lou, H.; Li, X.; Zhong, Z.; Sun, Z.; Chen, P.; Zhou, X.; Zuo, K.; Chen, Y.; Tang, X.; et al. Firered-ocr technical report. *arXiv* **2026**, arXiv:2603.01840.
23. Poznanski, J.; Soldaini, L.; Lo, K. olmOCR 2: Unit Test Rewards for Document OCR. *arXiv* **2025**, arXiv:2510.19817.
24. Zhang, J.; Liu, Y.; Wu, Z.; Pang, G.; Ye, Z.; Zhong, Y.; Ma, J.; Wei, T.; Xu, H.; Chen, W.; et al. MonkeyOCR v1.5 Technical Report: Unlocking Robust Document Parsing for Complex Patterns. *arXiv* **2025**, arXiv:2511.10390.

25. Yu, W.; Yang, Z.; Liu, Y.; Bai, X. DocThinker: Explainable Multimodal Large Language Models with Rule-based Reinforcement Learning for Document Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2025; pp. 837–847. [[CrossRef](#)]
26. Team, V.; Hong, W.; Yu, W.; Gu, X.; Wang, G.; Gan, G.; Tang, H.; Cheng, J.; Qi, J.; Ji, J.; et al. GLM-4.5V and GLM-4.1V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv* **2026**, arXiv:2507.01006.
27. Li, Z.; Liu, Y.; Liu, Q.; Ma, Z.; Zhang, Z.; Zhang, S.; Guo, Z.; Zhang, J.; Wang, X.; Bai, X. MonkeyOCR: Document Parsing with a Structure-Recognition-Relation Triplet Paradigm. *arXiv* **2025**, arXiv:2506.05218.
28. Zheng, H.; Li, Y.; Zhang, K.; Xin, L.; Zhao, G.; Liu, H.; Chen, J.; Lou, J.; Fu, Q.; Yang, R.; et al. Multimodal OCR: Parse Anything from Documents. *arXiv* **2026**, arXiv:2603.13032.
29. Fu, L.; Kuang, Z.; Song, J.; Huang, M.; Yang, B.; Li, Y.; Zhu, L.; Luo, Q.; Wang, X.; Lu, H.; et al. OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. In *Proceedings of the Advances in Neural Information Processing Systems*; Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., Chen, N., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2025; Volume 38, pp. 107930–107981. [[CrossRef](#)]
30. Chen, J.; Liu, X.; Wen, Z.; Wang, Y.; Huang, S.; Chen, H. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2026; pp. 3489–3499.
31. Zhang, X.; Zhu, Y.; Gu, C.; Yuan, X.; Zhao, Q.; Cao, J.; Tang, B.; Fan, S.; Shen, Y.; Shen, C.; et al. Hallucination Begins Where Saliency Drops. In *Proceedings of the International Conference on Learning Representations*; Vondrick, C., Hariharan, B., Raffel, C., Pinto, L., Yang, D., Faust, A., Eds.; OpenReview: Online, 2026; Volume 2026, pp. 15062–15083.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.